**SOLUCIONES ACTIVIDADES ADICIONALES TEMA 12**

Cree los siguientes subprogramas sobre la base de datos *Empresa*:

Depart (dept\_no, dnombre, loc)

Emple (emp\_no, apellido, oficio, dir, fecha\_alt, salario, comision, dept\_no)

1. Cree un procedimiento que reciba un número de departamento, nombre y localidad. Este procedimiento se ha de encargar de añadir un nuevo departamento a la tabla *Depart* con los datos recibidos como parámetros. En caso de que ya exista un departamento con el número recibido como parámetro, se deberá mostrar un mensaje de error significativo.

CREATE PROCEDURE AñadirDpto (numdep int, nomdep varchar(40), locdep varchar(40))

BEGIN

DECLARE duplicado bool default 0;

DECLARE CONTINUE HANDLER for 1062 SET duplicado = 1;

INSERT INTO Depart values (numdep, nomdep, locdep);

IF duplicado = 1 THEN

SELECT concat ('Ya existe un dpto. con el número ', numdep) ERROR;

ELSE

SELECT concat ('Añadido dpto. número ',numdep) Mensaje;

END IF;

END;

mysql> call AñadirDpto (50, 'FINANZAS', 'BURGOS'); //

+-------------------------+

| Mensaje |

+-------------------------+

| Añadido dpto. número 50 |

+-------------------------+

1 row in set (0.03 sec)

Query OK, 0 rows affected (0.03 sec)

mysql> call AñadirDpto (30, 'COMPRAS', 'LOGROÑO'); //

+-------------------------------------+

| ERROR |

+-------------------------------------+

| Ya existe un dpto. con el número 30 |

+-------------------------------------+

1 row in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

1. Cree un procedimiento que muestre para los dos empleados que más comisión cobran, su apellido, salario, comisión y el porcentaje que supone su comisión en relación con su salario.

CREATE PROCEDURE VerMasComisionados ()

BEGIN

DECLARE ape varchar(40);

DECLARE sal float;

DECLARE comi float;

DECLARE porcen float;

DECLARE numemple int default 0;

DECLARE fin bool default 0;

DECLARE c CURSOR for SELECT apellido, salario, comision,

round(comision/salario\*100,2)

FROM emple

ORDER BY comision desc;

DECLARE CONTINUE HANDLER FOR NOT FOUND SET fin = 1;

OPEN c;

FETCH c INTO ape, sal, comi, porcen;

WHILE fin = 0 and numemple < 2 DO

SELECT concat (ape,' Salario: ', sal,

' euros Comisión: ', comi,

' euros Porcentaje: ', porcen,'%')

"Datos empleado";

SET numemple = numemple + 1;

FETCH c INTO ape, sal, comi, porcen;

END WHILE;

CLOSE c;

END;

mysql> call VerMasComisionados();//

+--------------------------------------------------------------------------+

| Datos empleado |

+--------------------------------------------------------------------------+

| MARTÍN Salario: 965.00 euros Comisión: 1000.00 euros Porcentaje: 103.63% |

+--------------------------------------------------------------------------+

1 row in set (0.00 sec)

+----------------------------------------------------------------------+

| Datos empleado |

+----------------------------------------------------------------------+

| SALA Salario: 960.00 euros Comisión: 390.00 euros Porcentaje: 40.62% |

+----------------------------------------------------------------------+

1 row in set (0.01 sec)

Query OK, 0 rows affected (0.01 sec)

Cree los siguientes subprogramas sobre la base de datos *Pedidos*:

Pedido (RefPed, FecPed)

LíneaPedido (RefPed, CodArt, CantArt)

Artículo (CodArt, DesArt, PVPArt)

1. Cree un procedimiento que reciba un número y que muestre para los pedidos en que se solicite un número de artículos diferentes superior o igual al número recibido como parámetro, la referencia del pedido, su fecha, el número de artículos diferentes solicitados, el número de unidades de artículos solicitadas en total en el pedido y, por último, el importe del pedido.

CREATE PROCEDURE VerPedidosNArt(nartmin int)

BEGIN

DECLARE refe char(5);

DECLARE fec date ;

DECLARE nart int;

DECLARE nuni int;

DECLARE impped float;

DECLARE fin bool default 0;

DECLARE c CURSOR for SELECT P.RefPed, FecPed, count(\*),

sum(CantArt), sum(PVPArt \* CantArt)

FROM Pedido P, Articulo A, LineaPedido L

WHERE P.RefPed = L.RefPed AND L.CodArt = A.CodArt

GROUP BY P.RefPed, FecPed

HAVING count(\*) >= nartmin;

DECLARE CONTINUE HANDLER FOR NOT FOUND SET fin = 1;

OPEN c;

FETCH c INTO refe, fec, nart, nuni, impped;

WHILE fin = 0 DO

SELECT concat (refe,' ',fec,' ', nart,' artículos distintos '

,nuni, ' unidades',' ',impped, ' euros')

"Datos pedido";

FETCH c INTO refe, fec, nart, nuni, impped;

END WHILE;

CLOSE c;

END;

mysql> call VerPedidosNArt(2); //

+-----------------------------------------------------------------+

| Datos pedido |

+-----------------------------------------------------------------+

| P0001 2014-02-16 2 artículos distintos 22 unidades 20.40 euros |

+-----------------------------------------------------------------+

+-----------------------------------------------------------------+

| Datos pedido |

+-----------------------------------------------------------------+

| P0004 2014-02-25 3 artículos distintos 70 unidades 18.65 euros |

+-----------------------------------------------------------------+

Query OK, 0 rows affected (0.00 sec)

1. Cree un procedimiento que muestre para los tres artículos más caros de la base de datos, su código, descripción, precio y el número de pedidos en que está solicitado.

CREATE PROCEDURE VerArticulosCaros()

BEGIN

DECLARE codar char(5);

DECLARE descri varchar(30);

DECLARE precio float;

DECLARE numped int;

DECLARE numarts int DEFAULT 0;

DECLARE fin bool default 0;

DECLARE c CURSOR for SELECT A.CodArt, DesArt, PVPArt, count(\*)

FROM Articulo A, LineaPedido L

WHERE A.CodArt = L.CodArt

GROUP BY A.CodArt, DesArt, PVPArt

ORDER BY PVPArt DESC;

DECLARE CONTINUE HANDLER FOR NOT FOUND SET fin = 1;

OPEN c;

FETCH c INTO codar, descri, precio, numped;

WHILE fin = 0 and numarts<3 DO

SELECT concat (codar,' ',descri,' ', precio,' euros ', numped, ' pedido/s') "Datos artículo";

FETCH c INTO codar, descri, precio, numped;

SET numarts = numarts + 1;

END WHILE;

Close c;

END;

mysql> call VerArticulosCaros();//

+---------------------------------------------------+

| Datos artículo |

+---------------------------------------------------+

| A0078 Bolígrafo rojo normal 1.05 euros 1 pedido/s |

+---------------------------------------------------+

1 row in set (0.00 sec)

+--------------------------------------------+

| Datos artículo |

+--------------------------------------------+

| A0043 Bolígrafo azul 0.70 euros 3 pedido/s |

+--------------------------------------------+

1 row in set (0.00 sec)

+--------------------------------------+

| Datos artículo |

+--------------------------------------+

| A0075 Lápiz 2B 0.55 euros 1 pedido/s |

+--------------------------------------+

1 row in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Cree los siguientes subprogramas sobre la base de datos *Geografia*:

Localidades (id\_localidad, nombre, poblacion, n\_provincia)

Provincias (n\_provincia, nombre, superficie, id\_capital, id\_comunidad)

Comunidades (id\_comunidad, nombre, id\_capital)

1. Cree un procedimiento que reciba un número y que muestre para todas las localidades con más población que dicho número, el nombre de la localidad, el nombre de la provincia a la que pertenece y su población. Muestre al final un mensaje indicando: “Hay XX localidades con más de YYYY habitantes”.

create procedure ejer6repaso (num int)

begin

declare fin bool default 0;

declare nomloc varchar(50);

declare nomprov varchar(25);

declare pobla int;

declare numloc int default 0;

declare c cursor for select l.nombre, p.nombre, poblacion

from localidades l, provincias p

where l.n\_provincia = p.n\_provincia

and poblacion > num;

declare continue handler for not found set fin = 1;

open c;

fetch c into nomloc, nomprov, pobla;

while fin = 0 do

select concat ('Localidad: ',nomloc,' Provincia: ',nomprov,' ',pobla,' habitantes') Datos;

set numloc = numloc + 1;

fetch c into nomloc, nomprov, pobla;

end while;

close c;

select concat ('Hay ', numloc,' localidades con más de ',num,' habitantes') Mensaje;

end;

mysql> call ejer6repaso (250000)//

+------------------------------------------------------------+

| Datos |

+------------------------------------------------------------+

| Localidad: Coruña A Provincia: La Coruña 254822 habitantes |

+------------------------------------------------------------+

1 row in set (0.00 sec)

+---------------------------------------------------------+

| Datos |

+---------------------------------------------------------+

| Localidad: Vigo Provincia: Pontevedra 290582 habitantes |

+---------------------------------------------------------+

1 row in set (0.01 sec)

+--------------------------------------------------------+

| Datos |

+--------------------------------------------------------+

| Localidad: Bilbao Provincia: Vizcaya 370997 habitantes |

+--------------------------------------------------------+

1 row in set (0.01 sec)

+------------------------------------------------+

| Mensaje |

+------------------------------------------------+

| Hay 3 localidades con más de 250000 habitantes |

+------------------------------------------------+

1 row in set (0.01 sec)

Query OK, 0 rows affected (0.01 sec)

1. Cree un procedimiento que muestre los siguientes datos para las cinco localidades menos pobladas de la base de datos: su nombre, población y nombre de la provincia a la que pertenece la localidad.

create procedure LocalidadesMenosPobladas()

begin

declare fin bool default 0;

declare nomloc varchar(50);

declare pobl int;

declare prov varchar(25);

declare numloc int default 0;

declare c cursor for select l.nombre, poblacion, p.nombre

from localidades l, provincias p

where l.n\_provincia = p.n\_provincia

order by poblacion;

declare continue handler for not found set fin=1;

open c;

fetch c into nomloc, pobl, prov;

while fin = 0 and numloc < 5 do

select concat(nomloc,' ',pobl,' habitantes. Prov: ',prov) Datos;

set numloc = numloc + 1;

fetch c into nomloc, pobl, prov;

end while;

close c;

end;

mysql> call LocalidadesMenosPobladas();

+-----------------------------------------+

| Datos |

+-----------------------------------------+

| Villarroya 9 habitantes. Prov: La Rioja |

+-----------------------------------------+

1 row in set (0.03 sec)

+------------------------------------------------+

| Datos |

+------------------------------------------------+

| Torre en Cameros 11 habitantes. Prov: La Rioja |

+------------------------------------------------+

1 row in set (0.06 sec)

+---------------------------------------------+

| Datos |

+---------------------------------------------+

| Torremontalbo 15 habitantes. Prov: La Rioja |

+---------------------------------------------+

1 row in set (0.13 sec)

+---------------------------------------+

| Datos |

+---------------------------------------+

| Zarzosa 18 habitantes. Prov: La Rioja |

+---------------------------------------+

1 row in set (0.21 sec)

+-------------------------------------------+

| Datos |

+-------------------------------------------+

| Valdemadera 19 habitantes. Prov: La Rioja |

+-------------------------------------------+

1 row in set (0.24 sec)

Query OK, 0 rows affected (0.24 sec)